GPEM manuscript No.
(will be inserted by the editor)

VALIS: an Evolutionary Classification Algorithm

**Peter Karpov · Giovanni Squillero ·
Alberto Tonda**

Received: December 16, 2017/ Accepted: date

Abstract VALIS is an effective and robust classification algorithm with a focus on understandability. Its name stems from Vote-ALlocating Immune System as it evolves a population of artificial antibodies that can bind to the input data, and performs classification through a voting process. In the beginning of the training process, VALIS generates a set of random candidate antibodies; then, at each iteration it selects the best artificial antibodies to produce new candidates, while the less useful ones are discarded, and the process is iterated until a user-defined stopping condition. The evolutionary paradigm allows the user to get a visual insight into the learning dynamics, helping to supervise the process, pinpoint problems, and tweak feature engineering. VALIS is tested against nine state-of-the-art classification algorithms on six popular benchmark problems, proven to be competitive with all these well-established black-box techniques, and even superior for specific corner cases.

Keywords Evolutionary Machine Learning · Computational Intelligence · Artificial Immune Systems · Classifier System

Peter Karpov
Independent scholar
E-mail: PeterKarpov@inversed.ru

Alberto Tonda
UMR GMPA, INRA, AgroParisTech, Université Paris-Saclay
1 av. Brétignières, 78850, Thiverval-Grignon, France
Tel: +33 1 30 81 45 96
E-mail: alberto.tonda@inra.fr
ORCID: 0000-0001-5895-4809

Giovanni Squillero
Politecnico di Torino — DAUIN
Corso Duca degli Abruzzi 24, 10129, Torino, Italy
Tel: +39-011090.7186
E-mail: giovanni.squillero@polito.it
ORCID: 0000-0001-5784-6435

1 Introduction

Evolutionary machine-learning (EML) can be defined as a crossbreed between the fields of evolutionary computation (EC) and machine learning (ML). To avoid a blatant pleonasm, as the "obvious connection" between the processes of learning and evolution has been pointed out by Turing back in 1950 [40], the term is mostly used referring to the integration of well-established EC techniques and canonical ML frameworks. A first line of research ascribable to EML predates the recent ML windfall and focuses on using EC algorithms to optimize frameworks, it included remarkable studies in the 1990s, such as the attempts to determine optimal topologies for an artificial neural network using a genetic algorithm by Whitley, Starkweather and Bogart [44]. The other way around, a line tackling the use of ML techniques to boost EC algorithms appeared before 2000 [31]. Only more recently, scholars started proposing truly hybrid approaches — like the one described in this paper — where EC algorithms are deeply embedded in frameworks performing ML tasks.

Classification is one of the most studied argument in the ML community, with a considerable number of real-world applications and an impressive record of success stories. Classification consists in identifying the most probable class $\mathbf{y}$ an observation $\mathbf{x}$ belongs to, given a set of observations $\mathbf{Y_T}$ whose correct classification is known. Over the years, scholars developed a plethora of algorithms, each one being a different trade-off between conflicting goals, such as accuracy, precision, training speed, or even the mere understandability by human operators. After tackling a considerable number of case studies, it became apparent to the scientific community that there is no silver bullet, that is, no single classification algorithm is superior in all respects on all dataset, a conclusion consistent with the infamous *no free lunch* theorem [45].

We propose a classification algorithm named "VALIS" from *Vote-ALlocating Immune System*, whose core is based on a specific class of evolutionary algorithms known as artificial immune systems (AIS). Thanks to its distinctive evolutionary approach, the training outcomes of the the evolutionary core may be visualized and are easily understandable by the users, allowing to supervise the process, pinpoint problems, tweak the feature-engineering process accordingly, and even decide an early stop. The experimental evaluation clearly demonstrate its efficiency and remarkable robustness.

The rest of the paper is organized as follows: Section 2 summarizes the necessary concepts to better understand the scope of the current work; the proposed approach is detailed in Section 3, while the experimental evaluation is reported in Section 5; finally, Section 6 concludes the paper and drafts future perspectives.

2 Background

Correctly classifying new observations has a paramount practical usefulness, from *predicting* customers responses, to *detecting* fraudulent credit-card transactions. Classification has been tackled by scholars since the 1950s [3], but it is still one of the most actively studied argument in both the data mining and machine learning communities nowadays.

The process of classification relies on reliable data called *training set*, and or this reason, machine learning scholars call it a *supervised* technique. Conversely, the most notorious *unsupervised* technique is *clustering*, where observations are grouped together with no preconceived scheme.

Classifier systems may be broadly distinguished between *discriminative* and *generative*. The former try to determine a general rule starting from a set of direct mappings $\mathbf{x_i} \rightarrow \mathbf{y_i}$ from inputs to class; very popular classifiers such as *logistic regression* (LR) [14] and *support vector machines* (SVMs) [4] belong to this class. Differently, generative classifiers learn a model of the joint probability distributions $p(\mathbf{X}, \mathbf{Y})$ of the inputs and the classes, and make their predictions picking the most likely result; this category include *naïve Bayes* (NB) [46].

Another broad categorization discriminates between *lazy* and *eager* algorithms, where the first do not attempt to learn a decision rule or function, but rather calculate the response every time by analyzing all training data; *k-nearest neighbor* classifiers (kNN) [2] are an emblematic example of such an approach.

Nowadays, applications of classification span over quite different problem domains, such as text, multimedia, social networks, and biological data, and in a number of different scenarios, such as offline, streaming, and uncertain. Despite all the effort looking for versatile, efficient and robust classifier algorithms, no single solution has been found so far, and practitioners need to identify the optimal method for their specific problem, and possibly tweak its parameters.

AISs are inspired by immunology, and in particular by the *Immune Network* theory [23]. They emerged in the mid 1980s [16] and are now considered among the *bio-inspired* techniques, commonly classified as *evolutionary algorithms*. As there is no single universally adopted model, depending on which features of the immune system are modeled, AIS algorithms can be roughly split into four groups: negative selection [18], clonal selection [10,7], immune networks [9,38] and danger theory [19]. Recently, they have been successfully used for some machine learning tasks [42,41].

Negative selection is a process taking place in the thymus during which the cells that strongly bind to the "self" antigens are eliminated. In a similar way, the negative selection algorithms work by generating candidate detectors and eliminating those that match at least a single data sample from the self dataset. The resulting detector set can then be used to recognize the non-self data samples.

A viewpoint alternative to the self-nonself discrimination is advocated by the danger theory [29], according to which the immune system is sensitive to danger signals which are sent out by unnatural (as opposed to programmed) cell death. This idea is employed in the dendritic cell algorithm [19] which has been applied to the real-time anomaly detection.

Clonal selection theory [8] was proposed as a model of the acquired immune system. According to the clonal selection principle, lymphocytes that encounter a matching foreign antigen activate and proliferate to combat the intrusion. During the process, the antigen receptors are diversified as the corresponding genes undergo a very high rate of mutation. In addition to proliferation, lymphocytes can also differentiate into long lived memory cells. Algorithms from this class resemble the genetic algorithm, as they rely on selection, reproduction and mutation mechanisms.

The immune network theory argues that the immune system maintains an idiotypic network of interconnected $\mathcal{B}$ cells for antigen recognition, and that these cells both stimulate and suppress each other in certain ways that lead to the stabilization of the network; two $\mathcal{B}$ cells are connected if the affinities they share exceed a certain threshold, and the strength of the connection is directly proportional to the affinity they share. These ideas can be translated into a versatile and adaptive algorithm, somewhat similar to the one originally proposed by Holland [21].

AISs have been applied to solve real-world problems, including intrusion detection systems [17], credit card fraud detection [20], data mining [11], and the overall research in the field is steadily progressing [1]. While AISs have been widely exploited for unsupervised learning, such as clustering [35], far less applications tackled supervised learning or classification [36]. Among these, the best known are probably the Artificial Immune Recognition System (AIRS) [42] and CLONALG [7].

As AISs belong to the field of bio-inspired meta-heuristics, from which they take part of the nomenclature, it is worth to introduce some of the terminology that is going to be used in the following. A *generation* is an iteration of the algorithm; a common stop condition for bio-inspired heuristics is for the user to provide a maximum number of generations allowed. The set of all antibodies in the algorithm at a given generation is termed *population*. When antibodies are evaluated, their relative goodness is called *fitness*; by extension, the function used to evaluate them is usually referred to as *fitness function*. *Reproduction* is the name commonly assigned to the procedure of generating new solutions, starting from ones currently inside the population, usually through *mutation* (small random modifications inside of an antibody structure) or *crossover* (recombination of two or more antibodies). New solutions are generally termed *offspring*. At the end of each generation, a *replacement* procedure removes the worst-performing antibodies, in order to keep the population at the same initial size.

3 Proposed approach

Following the immunological metaphor of AISs, VALIS evolves a population of *antibodies*, while data samples represent *antigens*. *Molecules*, both antigens and antibodies, are specified by a number of parameters called their *generalized shapes*. The degree of interaction between an antibody b and an antigen g is quantified by a distance d_{bg} defined over the generalized shape space. The exact molecule representation and the definition of the distance function are problem-specific, but in most practical cases antibodies can be represented as spheres, and the Euclidean metric can be used to measure distances.

In a process similar to the immune response of a real immune system, VALIS select a specific class in response to a data sample. During the classification, each antibody bound to the input antigen votes using its class distribution, and the class with the maximum total votes is eventually returned as the classification result.

An early prototype of the system was tested on character recognition and source code classification problems [25]. The present one is a major improvement in terms of both accuracy and convergence speed, thanks to changes in the core algorithm, fitness calculation, learning rate schedule, and initialization.

The antibody's fitness value is calculated based on its individual local classification accuracy and a sharing factor, with the latter used to promote diversity

Algorithm 1 Training algorithm for VALIS

```
procedure TRAIN($\mathcal{G}$)
    $\mathcal{B} \leftarrow$ create $n_b$ antibodies
    $g \leftarrow 0$
    while $g < g_{\max}$ do
        evaluate($\mathcal{B}$)
        $\mathcal{O} \leftarrow$ create $n_o$ new antibodies
        remove $n_o$ worst-performing antibodies from $\mathcal{B}$
        $\mathcal{B} \leftarrow \mathcal{B} \cup \mathcal{O}$
    end while
end procedure
```

with a mechanism similar to *niching* [37]: if a large number of antibodies accumulates in the same area, the sharing factor will lower their fitness values and thus make other areas in the search space more attractive. This mechanism forces the system to cover the antigen space more uniformly. A straightforward generational scheme is used for training: each generation consists of selection, reproduction via crossover and mutation, and replacement dependent on the fitness value of each antibody.

The goal of VALIS is a collective problem solving, performed by the whole population. This is not the usual situation when EAs are used to optimize a function, but rather recall Holland's early works on *learning classifier systems* [21], or the more recent *cooperative co-evolution* algorithms [12,39]. The algorithms describing VALIS are reported in the following paragraphs, along with the necessary definitions. The complete training algorithm is reported in Algorithm 1.

3.1 Definitions

k a given class in the current classification task.

$\mathcal{K}$ the set of all classes: $\mathcal{K} = \{k_i\}$ with $i \in [0, n_\mathrm{k} - 1]$.

g an antigen, corresponding to a data sample $\mathbf{x}$; g_i is i-th data sample and it may be denoted as $\mathbf{x_i}$ in other papers tackling classifications; g^k denotes the class g belongs to.

n_g the number of antigens.

$\mathcal{G}$ the set of all antigens: $\mathcal{G} = \{g_i\}$ with $i \in [0, n_\mathrm{g} - 1]$.

b an antibody. In general, the representation of antibodies is problem-specific, but the present study adopts hypersphere-shaped antibodies defined by center b^c and radius b^r. Antibodies are the units of evolution; the fitness of the antibody is b^f.

n_b the number of antibodies.

$\mathcal{B}$ the set of all antibodies: $\mathcal{B} = \{b_i\}$ with $i \in [0, n_\mathrm{b} - 1]$.

n_f the number of features in the antigens.

α *abundance*, that is, the antibody-to-antigen ratio; it determines the degree of data reduction performed by the system, the easier the classification task is, the smaller this value can be. The value of $\alpha = 1$ (one antigen per antibody) has been used in all the reported experiments.

d_{bg} Euclidean distance from b^c, the center of antibody b, to antigen g.

w_{bg} The binding weight between antibody b and antigen g, that is, their degree of interaction.

$$w_{bg} = \mathrm{B}\left(\frac{d_{bg}}{b^{\mathrm{r}}}\right) \tag{1}$$

The binding function $\mathrm{B}(\cdot)$ can assume different forms; for most practical cases, a simple threshold can be used:

$$\mathrm{B}(x) = \begin{cases} 1 & \text{if } x \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

although other choices are possible in principle.

3.2 Core algorithm's steps

Data preprocessing — Since VALIS relies on the distance d_{bg} for classification, its performance is dependent on feature scaling. According to the nature of the data, various normalization strategies can be employed, or the preprocessing step can be skipped entirely. In the absence of prior knowledge, features should be normalized to unit variance. More advanced methods like metric learning [43] could be used to improve performance.

Antibody initialization — Create $n_{\mathrm{b}} = \lceil \alpha \cdot n_{\mathrm{g}} \rceil$ antibodies. Centers of the antibodies are randomly selected from the antigens, without replacement. The radius of each antibody is set to its distance from a random antigen. Other initialization methods are possible as well, but the impact of initialization on performance is outside of the scope of the present work.

Calculation of weight matrix — The matrix of binding weights $W = (w_{b_i g_j})$ between all antibodies and all antigens is calculated.

Calculation of class distributions — Let k be a specific class, $\mathcal{G}_k = \{g \in \mathcal{G} : g^{\mathrm{k}} = k\}$ is the set of all antigens belonging to the class k. defining $\bar{h}_{bk}$ as:

$$\bar{h}_{bk} = 1 + \sum_{g \in \mathcal{G}_k} w_{bg} \tag{3}$$

the distribution h_{bk} of classes of bound antigens for each antibody can be calculated with:

$$h_{bk} = \frac{\bar{h}_{bk}}{\sum_{k' \in \mathcal{K}} \bar{h}_{bk'}} \tag{4}$$

Class histograms are initialized with unit pseudo-counts; such a technique, known as *Laplacian smoothing*, has two important effects that reduce the risk of overfitting: fitness of antibodies bound to a very small number of antigens is effectively penalized due to the lower accuracy term; and such antibodies have a lesser impact during the voting process.

Fitness calculation — Fitness definition is the crucial part of VALIS, as it predominantly determines the dynamics of the system. The fitness $F(b)$ of a given antibody b is the weighted average of accuracies, adjusted for competition with other antibodies.

$$\mathrm{F}(b) = \frac{\sum_{g \in \mathcal{G}} \frac{(w_{bg})^2}{\sum_{b' \in \mathcal{B}} w_{b'g}} \cdot h_{bg^{\mathrm{k}}}}{\sum_{g \in \mathcal{G}} w_{bg}} \tag{5}$$

The term $h_{bg^{\mathrm{k}}}$ measures classification accuracy of antibody b, while the term $w_{bg}/\sum_{b' \in \mathcal{B}} w_{b'g}$ represents the sharing factor — the additional w_{bg} term comes from the weighted sum.

The sharing factor $w_{bg}/\sum_{b' \in \mathcal{B}} w_{b'g}$ is introduced to simulate competition for resources. Since the antigen g is shared, each antibody b receives only a portion of the total reward. Consequently, as soon as any particular area becomes overcrowded, the sharing factors drop, forcing the system to explore other regions. Without the sharing mechanism, all antibodies would converge to a single high accuracy area instead of covering the entire training set.

Accuracy calculation — Each antigen in the training set is classified via the vote allocation procedure. Each antibody casts its vote using its class distribution weighted by the binding weight. The vote casted to assign antigen $\bar{g}$ to class $\bar{k}$ is expressed by:

$$v_{gk} = \sum_{b \in \mathcal{B}} w_{bg} \cdot h_{bk} \tag{6}$$

The class $\mathcal{K}_S$ cumulating more votes is eventually chosen as the classification result for antigen g. If no antibody is bound to antigen g, that is $\forall b \in \mathcal{B} : w_{bg} = 0$, then the antibody with the lowest d_{bg}/b^{r} ratio is selected to cast a vote.

Parent selection — During the g-th generation, VALIS generate n_{o} new antibodies, by selecting n_{o} antibody pairs for reproduction with:

$$n_{\mathrm{o}} = \lceil n_{\mathrm{b}} \cdot \mathrm{L}\,(g) \rceil \tag{7}$$

A fixed offspring size cannot be entirely satisfactory: high values lead to large random fluctuation and poor accuracy, while low values result in slow convergence. Therefore an exponentially decaying size is adopted, with an effect similar a varying learning rate. Taking into account both the total number of antibodies n_{b} and the maximum number of generations $g_{\max}$ VALIS is evolving through:

$$\mathrm{L}(g) = \frac{1}{2} \cdot \left(\frac{2}{n_{\mathrm{b}}}\right)^{\frac{g}{g_{\max}}} \tag{8}$$

The probability for an antibody to be selected as a parent is directly proportional to its fitness. Such scheme results in faster equilibration of high-fitness areas and consequently faster convergence, and, contrary to the typical behavior in genetic algorithms, it does not seem to have a such a negative impact on antibody diversity — yet diversity is also enforced by the resource-sharing mechanism.

Reproduction — Create new antibodies via crossover and mutation operators. First, a *uniform crossover* is employed on antibody parameters, namely radius and center. Then, the new radius is mutated using a log-normal random multiplier, while the new center is mutated by adding a random variable with a log-uniform density; given the mutation probability p_{m}, the expected number of mutations steps is $e_{\mathrm{m}} = 1/p_{\mathrm{m}}$.

Replacement — Replace the n_{o} lowest-fitness antibodies with the newly generated offspring.

3.3 Comparison with other systems

Compared to AIRS and CLONALG, two well-known, established AIS-based algorithms, VALIS introduces significant novelties. First, in AIRS there is no difference between the representation of an antibody and an antigen, as both belong to the same generalized shape space. VALIS, on the other hand, allows for different representations, as long as the distance function is defined, which results in increased flexibility. Additionally, AIRS maintains a separate set of antibodies for each class and an extra set during training, while In VALIS, antibodies of various classes naturally coexist within a single population.

Finally, both AIRS and CLONALG exploit the k-nearest classification rule, which has no impact on the training phase; on the contrary, classification methods are an integral part of the training phase in VALIS. Independent voting based on binding weights is far more efficient — and it is far more biologically plausible: it relies on local antibody-antigen interactions, whilst the k-nearest rule requires sorting the antigens by distance. Antibodies of variable size have been employed in the V-detector [24], but being a variant of the negative selection algorithm, this solution can only perform binary self/non-self classification.

Not surprisingly, despite the different name of the underlying metaphor, Learning Classifier Systems, such as XCS and UCS [15], are quite similar to VALIS. The complete model is a population of local models, local model have accuracy-based fitness and their discovery is performed via genetic operators, and in case of multiple applicable models the result is determined by voting. However, there is a number of notable differences. The fitness sharing mechanism that plays a key role in VALIS is present in XCS, but not in UCS. LCS normally employ a subsumption mechanism to eliminate rules that are redundant or not general enough. In VALIS, the generality-accuracy tradeoff is achieved based on fitness definition alone.

Another difference is the use of class histograms instead of class labels. Overall, VALIS is a much simpler system, since it relies on self-organization driven only by fitness definition and requires no additional mechanisms: it features only a limited number of parameters (α, $g_{\max}$, plus eventually mutation rate and problem-specific constants employed by genetic operators), while for example XCS users need to set 20 parameters [27].

4 Visualizing experiments

An important feature of VALIS is the human-readable visualization of the training session. The state of the population can be depicted by selecting a specific plane and either projecting or dissecting the regions of space enclosed by the antibodies. When the goal is to evaluate the final result, the cross section method is found to produce more intelligible images (see Figure 3a for an example). Alternatively, when the goal is to tweak the learning process, antibody centers can be projected along with the antigens and the antibody-antigen bindings can be depicted by individual lines (see for example Figure 3b). The plane can be defined either by a pair of the original features or by two components resulting from the application of a dimensionality reduction algorithm. For instance, the top components obtained by using principal component analysis (PCA) [32,22], the t-distributed stochastic neighbor embedding (t-SNE) [28], or latent semantic analysis (LSA) [26].

Such visualization provides an additional insight into the system, as it allows to observe the training dynamics in details. A sample visualization of a training process is reported in Figure 1: colors have been selected to be easily separable, even by colorblind users. As the training process proceeds, the antibodies (colored circles) specialize in identifying several antigens (black dots) belonging to the same class; also, antibodies referring to the same classes start overlapping, reinforcing the predictions on well-identifiable antigens. If the classes are well separated in the chosen projection space, the visual effect is that - as iterations go on - the space will be cleanly separated into regions with different colors. Antigens close to the threshold between two classes can be identified as being inside several circles of different colors. This kind of visualization makes it easy to identify issues in the training process, shown as great overlapping in the circles describing antibodies.

The first type of visualization can be straightforwardly compared to the second, showing also antibodies and depicting bindings antigen-antibody as lines, as presented in the left and right columns of Figure 3. This second depiction makes it easier for the user to correctly assess population dimensionality, as the number of antibodies needed to cover all samples of a certain class is made evident by the density of lines in various colors. Figure 4 reports a comparison of the two visualization approaches on a different dataset.

Additional examples of visualizations can be found on the project homepage[1].

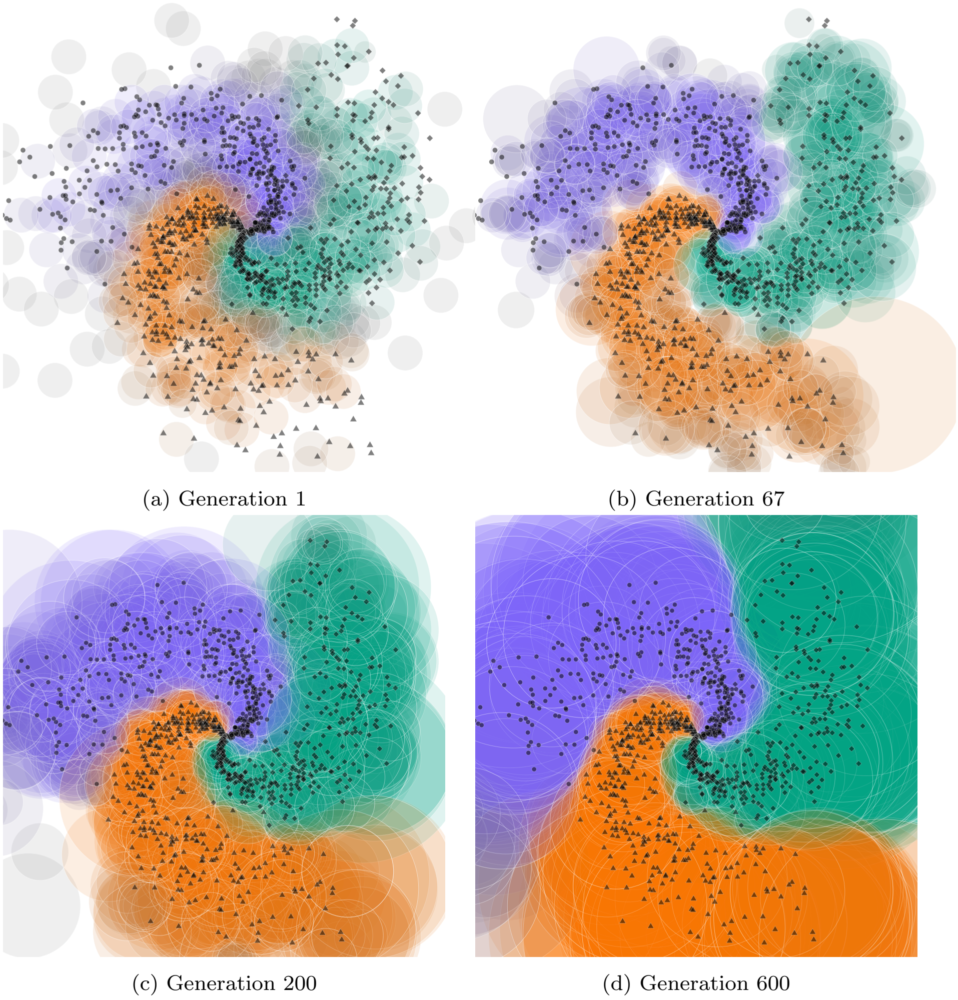
5 Experimental Evaluation

In order to assess VALIS, we compare its performance against 9 state-of-the-art classifiers implemented in the Python package `scikit-learn` [33]: AB (Adaptive Boost) [47], CART (Decision Tree) [5], kNN (k-Nearest Neighbors) [2], LR (Logistic Regression) [14], LDA (Linear Discriminant Analysis) [34], NB (Naive Bayes) [46], QDA (Quadratic Discriminant Analysis) [13], RF (Random Forest) [6], SVM (Support Vector Machines) [4].

Six benchmark datasets taken from the UCI Machine Learning Repository[2] and the MASS package are used as case studies: *Crabs*, *Glass*, *Ionosphere*, *Iris*, *Sonar*

[1] `http://inversed.ru/AIS.htm`

[2] `http://archive.ics.uci.edu/ml/`


(a) Generation 1

(b) Generation 67

(c) Generation 200

(d) Generation 600

Fig. 1: Visualization of VALIS' training process on a synthetic problem with two features and three classes, at selected generations. Antigens (data points) are represented as black dots, with class membership denoted by their shape. Antibodies are represented as circles with colors linked to their class distributions. Population was initialized with random positions and fixed radii. Learning rate was held constant throughout the training process.

and *Wine*. Basic dataset information is summarized in Table 1. All dataset features are normalized to zero mean and unit variance prior to running the experiments. Default settings are used for all classifiers, except kNN, for which we present results for $k = 1, 3, 5, 7$. VALIS' parameters are set as follows: $\alpha = 1$, $g_{\max} = 600$. The results are obtained by averaging 5 10-fold cross-validation runs with randomized splits. Complete results are reported in table 2, while a Principal Component

Analysis visualization of the training procedure on the *Iris* and *Wine* datasets are depicted in Figure 3 and Figure 4, respectively.

Table 1: Basic dataset information.

	Samples	**Variables**	**Classes**	**Source**
Iris	150	4	3	UCI
Wine	178	13	3	UCI
Glass	214	9	6	UCI
Sonar	208	60	2	UCI
Ionosphere	351	34	2	UCI
Crabs	200	5	4	MASS

Table 2: VALIS' performance, compared against state-of-the-art classifiers implemented in the Python package `scikit-learn` [33], on six datasets, using 6 repetitions of a 10-fold cross validation. For each classifier and dataset, the cross-validation accuracy is reported, along with the standard deviation (between parentheses), results better than VALIS are shown in **bold**.

	Crabs	**Glass**	**Ionosphere**	**Iris**	**Sonar**	**Wine**
AB	0.593 (0.023)	0.446 (0.024)	0.926 (0.008)	0.943 (0.010)	**0.820** (0.021)	0.894 (0.024)
CART	0.736 (0.023)	0.663 (0.022)	0.883 (0.008)	0.950 (0.007)	0.718 (0.021)	0.900 (0.013)
kNN (k=1)	**0.885** (0.011)	**0.700** (0.010)	0.863 (0.006)	0.944 (0.004)	**0.869** (0.008)	0.954 (0.005)
kNN (k=3)	0.823 (0.015)	**0.706** (0.012)	0.842 (0.005)	0.944 (0.005)	**0.858** (0.014)	0.956 (0.006)
kNN (k=5)	0.816 (0.012)	0.653 (0.012)	0.845 (0.005)	0.951 (0.004)	0.815 (0.013)	0.966 (0.005)
kNN (k=7)	0.786 (0.013)	0.639 (0.015)	0.836 (0.004)	0.955 (0.005)	0.800 (0.013)	0.965 (0.007)
LR	**0.918** (0.006)	0.618 (0.016)	0.884 (0.006)	0.897 (0.009)	0.771 (0.014)	**0.984** (0.004)
LDA	**0.946** (0.005)	0.623 (0.018)	0.866 (0.005)	**0.979** (0.002)	0.745 (0.021)	**0.986** (0.004)
NB	0.373 (0.013)	0.465 (0.014)	0.886 (0.004)	**0.953** (0.003)	0.680 (0.008)	**0.974** (0.004)
QDA	**0.939** (0.005)	0.129 (0.037)	0.908 (0.004)	**0.972** (0.004)	0.742 (0.030)	**0.991** (0.005)
RF	0.773 (0.024)	**0.743** (0.020)	0.923 (0.007)	0.951 (0.007)	0.780 (0.024)	**0.979** (0.007)
SVM	0.809 (0.012)	**0.701** (0.011)	**0.940** (0.003)	**0.964** (0.005)	**0.841** (0.012)	**0.983** (0.003)
VALIS	0.876 (0.011)	0.689 (0.024)	0.928 (0.007)	0.956 (0.005)	0.818 (0.020)	0.972 (0.005)

Accuracies relative to the best performing algorithm were then calculated for each dataset. The results indicate that VALIS is a robust and efficient classifier: although it is never the most performing for any particular problem, it ranks first both by geometric mean and minimum of relative accuracy; the margins, however, are not statistically significant, see Table 3. A radar-plot for the accuracies of all considered classifiers is reported in Figure 2. It it easy to notice that VALIS has a balanced performance across all 6 datasets. In contrast, LDA, QDA and LR demonstrate excellent accuracy on certain problems but underperform or completely fail on others.

Obtaining a fair evaluation of VALIS against popular AIS classifiers is not trivial, as the code for AIRS and CLONALG is not freely available. However, in Table 4 we present the results obtained by VALIS on the same datasets used in the AIRS publication [30], under the same conditions. The table also presents the comparison of VALIS against XCS and UCS with the data taken from [15], although the experimental conditions were not strictly identical as 10-fold stratified cross-validation is employed in LCS tests.

6 Conclusions

In this paper, we presented VALIS, a novel immune-inspired supervised learning algorithm. Compared to other Artificial Immune Systems, VALIS differs in terms of the population structure and dynamics, as the antibodies related to different classes coexist and compete within a single population. From the algorithmic point of view, proposed approach is in fact more similar to learning classifier systems. In experiments conducted on six popular benchmark problems, VALIS performed on par or better than several established classification algorithms.

Remarkably, the system exhibits emergent global behavior as the result of local antibody interactions. Although the training is based on individual fitness, the population as a whole converges towards a higher collective classification accuracy. Since training relies on self-organization of the antibody population, the algorithm is simple and has few parameters.

Exploring alternative antibody definitions is a promising direction for future work. By defining an appropriate antibody representation, VALIS can be adapted to other types of problems including ones with discrete or nominal features. Further experiments would also be required to better assess the effects of various crossover and mutation operators on the performance.

Table 3: Geometric means and minimums of relative accuracies, sorted in decreasing performances.

	GM	**MIN**
VALIS	0.956 (0.009)	0.926 (0.013)
SVM	0.956 (0.006)	0.855 (0.013)
kNN (k=1)	0.953 (0.006)	0.918 (0.007)
RF	0.940 (0.007)	0.817 (0.025)
kNN (k=3)	0.938 (0.007)	0.870 (0.017)
LDA	0.933 (0.008)	0.838 (0.033)
LR	0.922 (0.007)	0.832 (0.031)
kNN (k=5)	0.920 (0.006)	0.863 (0.013)
kNN (k=7)	0.907 (0.007)	0.831 (0.014)
CART	0.883 (0.009)	0.778 (0.025)
AB	0.820 (0.011)	0.600 (0.036)
NB	0.747 (0.007)	0.394 (0.014)
QDA	0.722 (0.035)	0.174 (0.050)

Table 4: Comparison with with AIRS's accuracies as reported in [30], and with LCS's, as reported in [15]. Results better than VALIS are shown in **bold**.

	Glass	**Ionosphere**	**Iris**	**Sonar**	**Wine**
AIRS-1	*n.a.*	0.869	**0.960**	**0.841**	*n.a.*
AIRS-7	*n.a.*	0.886	0.953	0.765	*n.a.*
XCS	**0.708**	*n.a.*	0.947	*n.a.*	0.951
UCS	**0.708**	*n.a.*	0.947	*n.a.*	0.972
VALIS	0.689	0.928	0.956	0.818	0.972

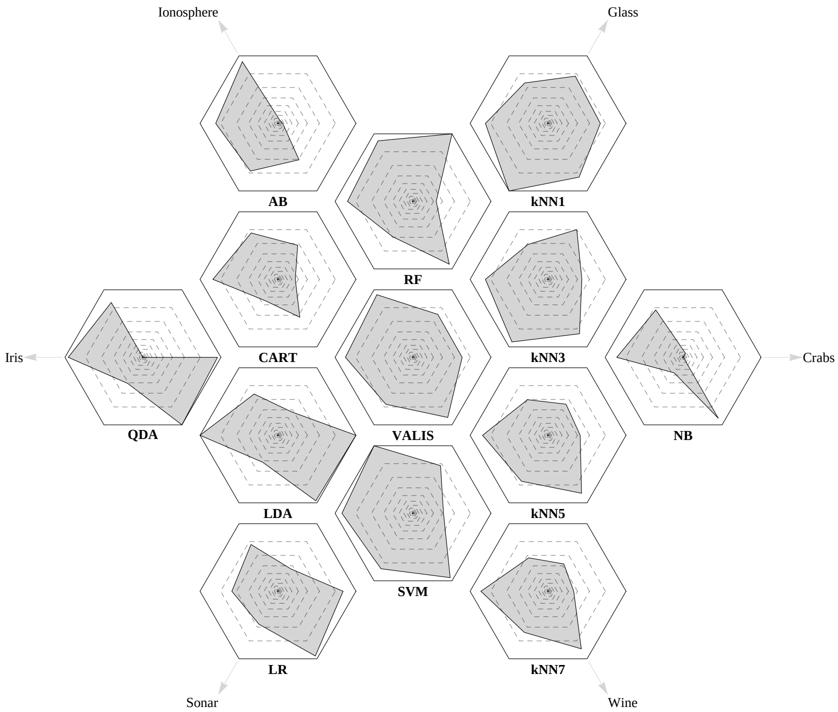


Fig. 2: Radar plot for the relative accuracies of classifiers included in the comparisons, on the six considered benchmarks. Dashed lines correspond to 5% increments.

Acknowledgements Peter Karpov would like to thank Anton Shepelev, Brian Bucklew, Neil J. A. Sloane and Glenn Fiedler for financial support through his Patreon page[3].

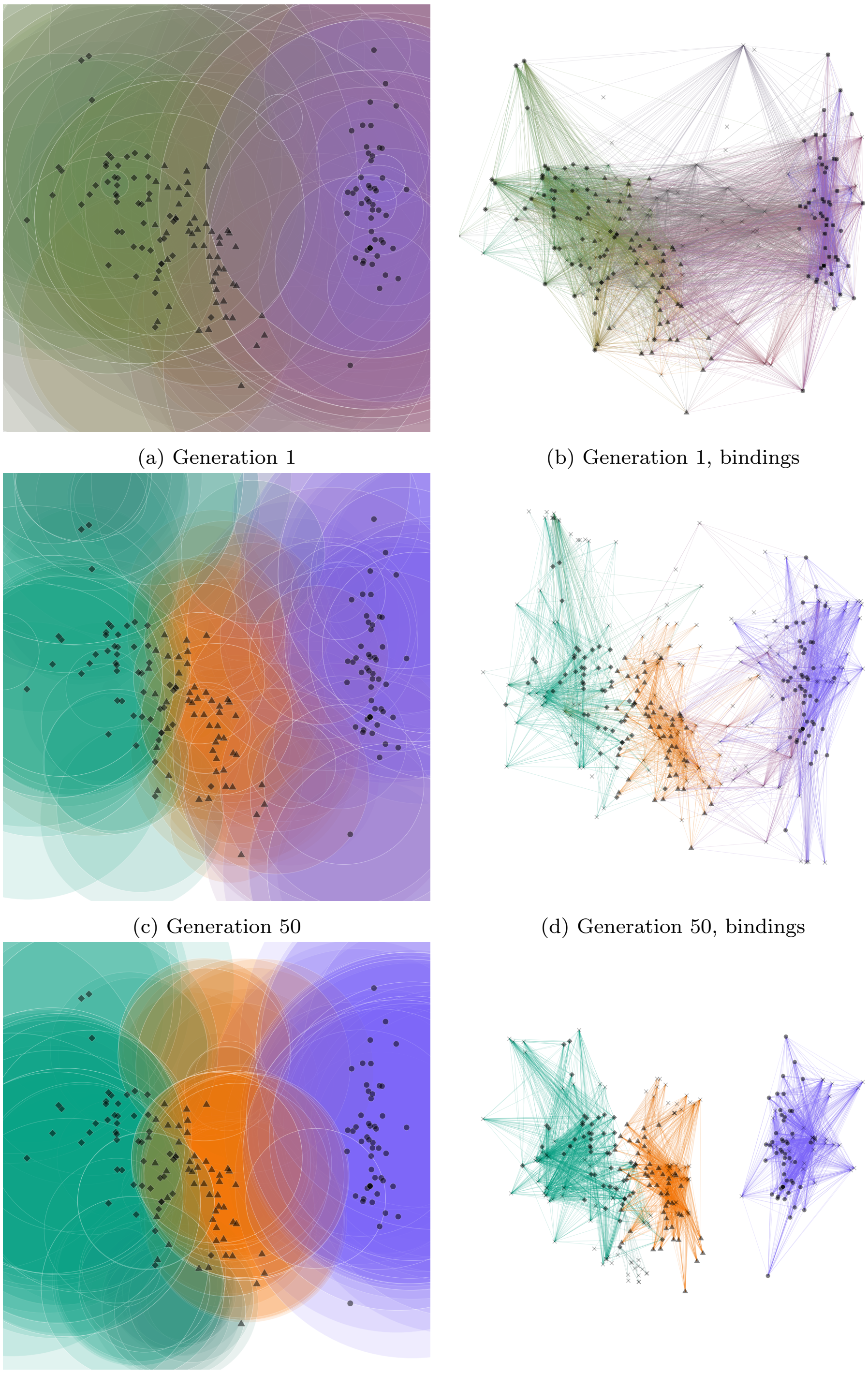
References

1. Aickelin, U., Dasgupta, D., Gu, F.: Artificial immune systems. In: Search Methodologies, pp. 187–211. Springer (2014)
2. Altman, N.S.: An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician **46**(3), 175–185 (1992)
3. Anderson, T.W.: Classification by multivariate analysis. Psychometrika **16**(1), 31–50 (1951). DOI 10.1007/bf02313425. URL https://doi.org/10.1007%2Fbf02313425
4. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on Computational learning theory, pp. 144–152. ACM (1992)
5. Breiman, L.: Pasting small votes for classification in large databases and on-line. Machine Learning **36**(1-2), 85–103 (1999)
6. Breiman, L.: Random forests. Machine learning **45**(1), 5–32 (2001)
7. Brownlee, J.: Clonal selection theory & CLONALG – the clonal selection classification algorithm (CSCA). Swinburne University of Technology (2005)
8. Burnet, F.M.: The clonal selection theory of acquired immunity. Nashville, Vanderbilt University Press (1959)

[3] https://www.patreon.com/inversed

9. de Castro, L.N., Von Zuben, F.J.: ainet: an artificial immune network for data analysis. Data mining: a heuristic approach **1**, 231–259 (2001)
10. de Castro, L.N., Zuben, F.J.V.: Learning and optimization using the clonal selection principle. IEEE Transactions on Evolutionary Computation **6**(3), 239–251 (2002). DOI 10.1109/TEVC.2002.1011539
11. Cayzer, S., Aickelin, U.: On the effects of idiotypic interactions for recommendation communities in artificial immune systems. Tech. rep. (2008)
12. Collet, P., Lutton, E., Raynal, F., Schoenauer, M.: Polar IFS+ parisian genetic programming= efficient ifs inverse problem solving. Genetic Programming and Evolvable Machines **1**(4), 339–361 (2000)
13. Cover, T.M.: Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. IEEE transactions on electronic computers (3), 326–334 (1965)
14. Cox, D.R.: The regression analysis of binary sequences. Journal of the Royal Statistical Society. Series B (Methodological) pp. 215–242 (1958)
15. Ester Bernadò-Mansilla, J.M.G.G.: Accuracy-based learning classifier systems: Models, analysis and applications to classification tasks. Evolutionary Computation **11**(3), 209–238 (2003)
16. Farmer, J., Packard, N.H., Perelson, A.S.: The immune system, adaptation, and machine learning. Physica D: Nonlinear Phenomena **22**(1-3), 187–204 (1986). DOI 10.1016/0167-2789(86)90240-x. URL `https://doi.org/10.1016%2F0167-2789%2886%2990240-x`
17. Forrest, S., Hofmeyr, S.A., Somayaji, A.: Computer immunology. Communications of the ACM **40**(10), 88–96 (1997)
18. Forrest, S., Perelson, A.S., Allen, L., Cherukuri, R.: Self-nonself discrimination in a computer. In: Proceedings of the 1994 IEEE Symposium on Security and Privacy, SP '94, pp. 202–. IEEE Computer Society, Washington, DC, USA (1994)
19. Greensmith, J., Aickelin, U., Cayzer, S.: Detecting Danger: The Dendritic Cell Algorithm, pp. 89–112. Springer London, London (2008). DOI 10.1007/978-1-84800-261-6_5
20. Halvaiee, N.S., Akbari, M.K.: A novel model for credit card fraud detection using artificial immune systems. Applied Soft Computing **24**, 40–49 (2014)
21. Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. The University of Michigan Press (1975). URL `http://books.google.it/books?id=YE5RAAAAMAAJ`
22. Hotelling, H.: Analysis of a complex of statistical variables into principal components. Journal of educational psychology **24**(6), 417 (1933)
23. Jerne, N.K.: Towards a network theory of the immune system. In: Annales d'immunologie, vol. 125, pp. 373–389 (1974)
24. Ji, Z., Dasgupta, D.: Real-valued negative selection algorithm with variable-sized detectors. In: Genetic and Evolutionary Computation Conference, pp. 287–298. Springer (2004)
25. Karpov, P.M.: VALIS, a novel immune-inspired classification algorithm. In: Proceedings of the 12th national conference on artificial intelligence with international participation CAI-2010, vol. 2, pp. 325–334. Fizmatlit (2010)
26. Landauer, T.K., Foltz, P.W., Laham, D.: An introduction to latent semantic analysis. Discourse processes **25**(2-3), 259–284 (1998)
27. M. V. Butz, S.W.W.: An algorithmic description of xcs. Soft Computing **6**(3), 144–153 (2002)
28. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research **9**(Nov), 2579–2605 (2008)
29. Matzinger, P.: Tolerance, danger, and the extended family. Annual Review of Immunology **12**(1), 991–1045 (1994)
30. Meng, L., van der Putten, P., Wang, H.: A comprehensive benchmark of the artificial immune recognition system (airs). In: Proceedings of the First International Conference on Advanced Data Mining and Applications, ADMA'05, pp. 575–582. Springer-Verlag, Berlin, Heidelberg (2005). DOI 10.1007/11527503_68
31. Michalski, R.S.: Learnable evolution model: Evolutionary processes guided by machine learning. Machine Learning **38**(1), 9–40 (2000)
32. Pearson, K.: Liii. on lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **2**(11), 559–572 (1901)

33. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
34. Rao, C.R.: The utilization of multiple measurements in problems of biological classification. Journal of the Royal Statistical Society. Series B (Methodological) **10**(2), 159–203 (1948)
35. Shang, R., Qi, L., Jiao, L., Stolkin, R., Li, Y.: Change detection in SAR images by artificial immune multi-objective clustering. Engineering Applications of Artificial Intelligence **31**, 53–67 (2014)
36. Sotiropoulos, D.N., Tsihrintzis, G.A.: Experimental evaluation of artificial immune system-based learning algorithms. In: Machine Learning Paradigms, pp. 237–323. Springer (2017)
37. Squillero, G., Tonda, A.: Divergence of character and premature convergence: A survey of methodologies for promoting diversity in evolutionary optimization. Information Sciences **329**, 782–799 (2016)
38. Timmis, J., Neal, M., Hunt, J.: An artificial immune system for data analysis. Biosystems **55**(1), 143–150 (2000)
39. Tonda, A., Lutton, E., Squillero, G.: A benchmark for cooperative coevolution. Memetic Computing **4**(4), 263–277 (2012). DOI 10.1007/s12293-012-0095-x. URL `http://porto.polito.it/2503591/`
40. Turing, A.M.: Computing machinery and intelligence. Mind **59**(236), 433–460 (1950)
41. Ulutas, B.H., Kulturel-Konak, S.: A review of clonal selection algorithm and its applications. Artificial Intelligence Review **36**(2), 117–138 (2011)
42. Watkins, A., Timmis, J., Boggess, L.: Artificial immune recognition system (airs): An immune-inspired supervised learning algorithm. Genetic Programming and Evolvable Machines **5**(3), 291–317 (2004)
43. Weinberger, K.Q., Blitzer, J., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. In: Y. Weiss, P.B. Schölkopf, J.C. Platt (eds.) Advances in Neural Information Processing Systems 18, pp. 1473–1480. MIT Press (2006)
44. Whitley, D., Starkweather, T., Bogart, C.: Genetic algorithms and neural networks: Optimizing connections and connectivity. Parallel computing **14**(3), 347–361 (1990)
45. Wolpert, D.H.: The lack of a priori distinctions between learning algorithms. Neural computation **8**(7), 1341–1390 (1996)
46. Zhang, H.: The optimality of naive bayes. AA **1**(2), 3 (2004)
47. Zhu, J., Zou, H., Rosset, S., Hastie, T.: Multi-class adaboost. Statistics and its Interface **2**(3), 349–360 (2009)



(a) Generation 1

(b) Generation 1, bindings

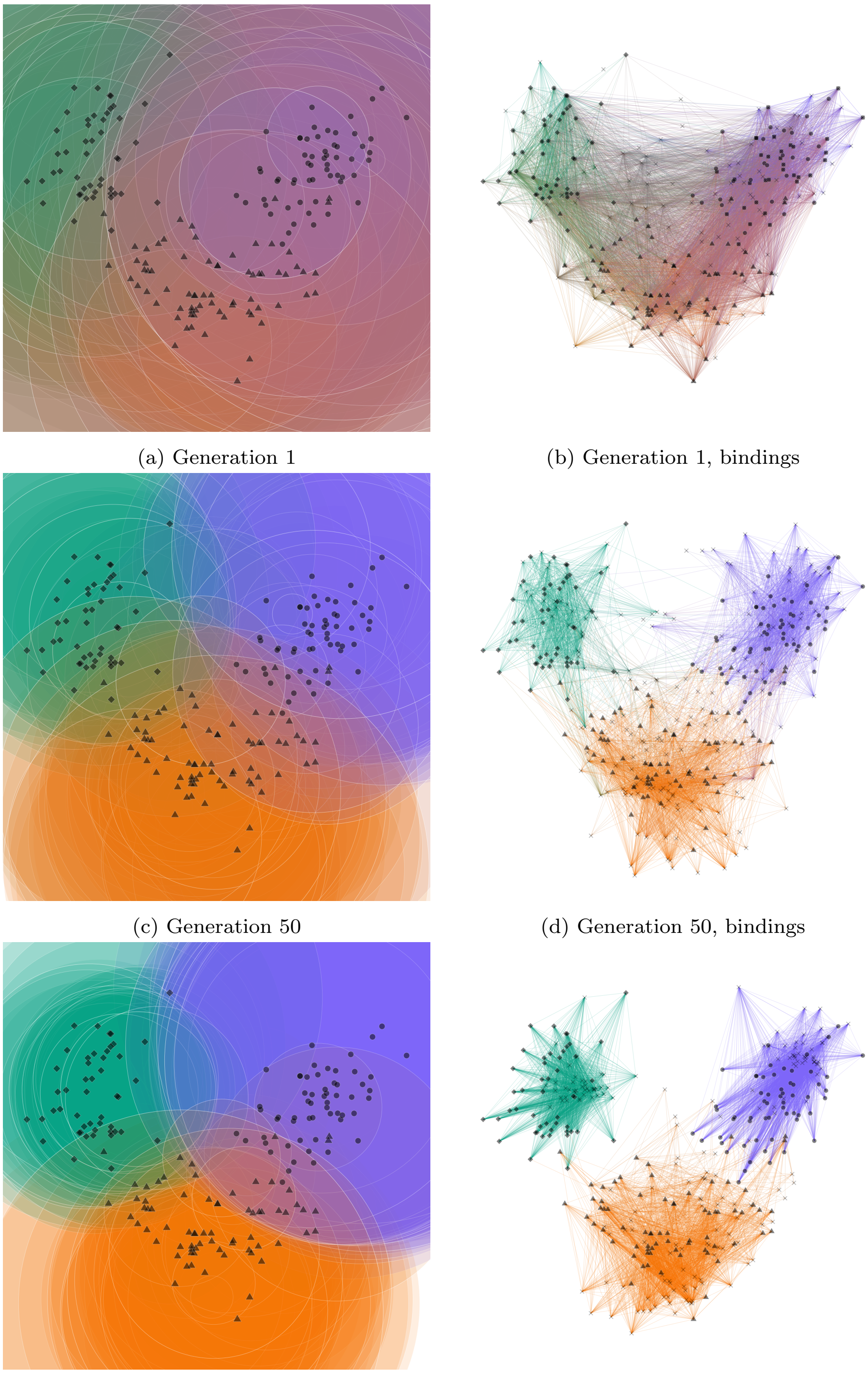
(c) Generation 50

(d) Generation 50, bindings

(e) Generation 600

(f) Generation 600, bindings

Fig. 3: Visualization of VALIS' training process on the Iris dataset, using Principal Component Analysis, at selected generations. Left: antibody cross sections, right: antibody - antigen bindings. Black dots depict antigens (data points), with class membership denoted by shape (circle, triangle, diamond for classes 1, 2, 3, respectively), and crosses depict antibody centers. Colors represent antibody class distributions (green, orange, violet, for classes 1, 2, 3, respectively). Parameters are described in Section 5.



(a) Generation 1

(b) Generation 1, bindings

(c) Generation 50

(d) Generation 50, bindings

(e) Generation 600

(f) Generation 600, bindings

Fig. 4: Visualization of VALIS' training process on the Wine dataset, using Principal Component Analysis, at selected generations. Left: antibody cross sections, right: antibody - antigen bindings. Black dots depict antigens (data points), with class membership denoted by shape (circle, triangle, diamond for classes 1, 2, 3, respectively), and crosses depict antibody centers. Colors represent antibody class distributions (green, orange, violet, for classes 1, 2, 3, respectively). Parameters are described in Section 5.